Model Context Protocol

Antes de intentar consumir este servicio debe indicar en las cabeceras de la petición (**headers**) los parámetros de autenticación para identificarse ante el sistema. Use la llave **username** para indicar el nombre de usuario (con el mismo que inicia sesión en la interfaz web y **password** para indicar su contraseña.

Algunas consultas deben tener las siguientes cabeceras. Más abajo en la tabla de operaciones se indican cuales de ellas son necesarias dependiendo de la operación.

Header	Valor	Explicación
endpoint	jason_and716_agt	Identificación del endpoint
desde	2022-01-01	Rango de fecha "desde"
hasta	2022-02-28	Rango de fecha "hasta"

Las operaciones disponibles son:

Operación	Método	Headers	URL Ejemplo
Obtener el resumen general de las alertas	**GET**	**username, password, desde, hasta**	https://url/rest/mcp?general
Obtener una vista detallada de las alertas de un endpoint	**GET**	**username, password, endpoint, desde, hasta**	https://url/rest/mcp?detalle

Puede configurar su LLM para crear dos herramientas (tools) en Python que consuman este servicio API. Este script en Python implementa las dos herramientas (vista general y vista detallada de alertas):

```
from mcp.server.fastmcp import FastMCP
from datetime import datetime, timedelta
import httpx
import json

params = json.load(open('config.json'))

mcp = FastMCP("NOFRAUD")

#
```

```
# Descripcion de la herramienta: hace un resumen de todas las alertas que existen en un rango de fechas
#

@mcp.tool()
async def visionGeneral(desde: str, hasta: str) -> str:
    """
    Hace un resumen general de todas las alertas que existen en un rango de fechas.
    Esta herramienta solo se debe usar cuando se solicita un resumen general, no cuando se solicitan detalles.
    Cuando se solicitan detalles se debe usar la otra herramienta 'visionDetallada'.

    Estos son algunos ejemplos para usar esta herramienta:

    1. Podrias hacerme un resumen de las alertas desde el 1 de Abril al 30 de Abril de 2025?:
       visionGeneral(2025-04-01, 2025-04-30).
    2. Me podrias indicar si entre el periodo del 1 de Enero al 31 de Marzo de 2025 existieron alertas?:
       visionGeneral(2025-01-01, 2025-03-31).

    Requisitos para el uso de esta herramienta:

    1. La fecha de inicio debe ser anterior a la fecha de fin.
    2. Las fechas deben ser en formato YYYY-MM-DD.
    """
    global params
    async with httpx.AsyncClient() as client:
        url = params["url"] + "?general"
        headers = {
            "Content-Type": "application/json",
            "username": params["username"],
            "password": params["password"],
            "desde": str(desde),
            "hasta": str(hasta),
        }
        response = await client.get(url, headers=headers)
        response.raise_for_status()

    return response.json()

@mcp.tool()
async def visionDetallada(endpoint: str, desde: str, hasta: str) -> str:
    """
```

```
    Detalla las alertas que tiene un endpoint en especifico en un rango de fechas.
    Esta herramienta solo se debe usar cuando se solicitan detalles, no cuando se solicita un resumen.
    Cuando se solicita un resumen se debe usar la herramienta 'visionGeneral'.

    Estos son algunos ejemplos para usar esta herramienta:

    1. Podrias mostrarme las alertas del endpoint 'jhondoe' del 1 de Abril al 30 de Abril de 2025?:
       visionDetallada('jhondoe', 2025-04-01, 2025-04-30).
    2. Podrias detallarme las alertas del endpoint 'jhondoe@nofraud.la' del 1 de Enero al 31 de Marzo de 2025?:
       visionDetallada('jhondoe@nofraud.la', 2025-01-01, 2025-03-31).

    Requisitos para el uso de esta herramienta:

    1. El endpoint debe ser un nombre de usuario o un email. No se acepta 'todos' como endpoint.
    2. La fecha de inicio debe ser anterior a la fecha de fin.
    3. Las fechas deben ser en formato YYYY-MM-DD.
    """
    global params
    async with httpx.AsyncClient() as client:
        url = params["url"] + "?detalle"
        headers = {
            "Content-Type": "application/json",
            "username": params["username"],
            "password": params["password"],
            "endpoint": endpoint,
            "desde": str(desde),
            "hasta": str(hasta),
        }
        response = await client.get(url, headers=headers)
        response.raise_for_status()

    return response.json()

if __name__ == "__main__":
    mcp.run()
```

El archivo de configuración que debe crear según su instancia de The Fraud Explorer:

```
{
    "username": "mcp-username",
```

```
    "password": "su_password_mcp",
    "url": "https://demo.thefraudexplorer.com/rest/mcp"
}
```

The Fraud Explorer es un software que junto con **FraudGPT** detecta el fraude y la corrupción en las organizaciones.

Revision #2
Created 23 July 2025 03:18:40 by Julian Rios
Updated 25 July 2025 22:04:02 by Julian Rios